EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L6	61	714/5.ccls. and @pd>="20070106"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:33
L7	161	714/6.ccls. and @pd>="20070106"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 12:58
L8	54	711/161.ccls. and @pd>="20070106"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 12:59
L9	232	711/162.ccls. and @pd>="20070106"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:02
L10	49	(CHANDRASEKARAN-SASHIKANTH KEHOE-WILLIAM-F).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:00
L11	236	707/204.ccls. and @pd>="20070106"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:07
L12	223	714/16.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:37
L13	1025	714/15.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:52
L14	36	714/15.ccls. and (redo and transaction)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:40

L15	4560	709/219.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:38
L16	9	709/219.ccls. and (redo and transaction)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:38
L17	102	714/19.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:40
L18	357	714/20.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:40
L19	36	714/20.ccls. and (redo and transaction)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:40
L20	145	mirror and redo and transaction	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/07/06 13:52
S1	2	"6081411".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/06 10:24
S2	2	10/759894	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/06 10:24
S3	1679	707/204.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/06 11:28

EAST Search History

S4	118	S3 AND (synchronous AND asynchronous)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/06 11:29
S5	1901	711/161,162.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/09 15:29
S6	1679	707/204.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/09 15:29
S7	358	S5 AND S6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/09 15:29
S8	3222	S5 OR S6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/09 15:33
S9	323	S8 AND (synchronous$2 SAME asynchronous$2)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/09 16:02
S10	285	S8 AND (synchronous$2 WITH asynchronous$2)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/09 16:02
S11	11	S8 AND (synchronous$2 WITH asynchronous$2 WITH switch$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/09 16:02
S12	1	("2004/0024979").URPN.	USPAT	OR	ON	2006/10/11 10:39

S13	11	(("0530855") or ("5917998") or ("20020083036") or ("6505307") or ("20030188114") or ("6643795") or ("6144999") or ("6163856") or ("6757790") or ("200300790019") or ("20030187861")).PN.	US-PGPUB; USPAT; USOCR; FPRS	OR	OFF	2006/10/11 10:41
S14	646	((synchronous$2 NEAR3 asynchronous$2) WITH (mode$1 OR switch$3)).AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40
S15	438	((synchronous$2 NEAR3 asynchronous$2) NEAR2 (mode$1 OR switch$3)).AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:39
S16	1908	711/161,162.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40
S17	1686	707/204.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40
S18	363	S16 AND S17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40
S19	0	S14 AND S18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40
S20	3229	((synchronous$2 NEAR3 asynchronous$2) WITH (mode$1 OR switch$3))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40

EAST Search History

S21	9	S20 AND S18	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40
S22	3231	S16 OR S17	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40
S23	11	S22 AND (synchronous$2 WITH asynchronous$2 WITH switch$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:40
S24	9	S21 NOT S23	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:41
S25	9	S21 NOT S23	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:58
S26	2804	714/5,6.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 14:58
S27	145	S26 AND bitmap	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/11 15:01
S28	10	S26 AND bitmap.AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/12 10:22

S29	34	differential ADJ bitmap	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/12 11:07
S30	167	((meta ADJ data) OR metadata) WITH synchronous$2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/13 09:15
S31	1910	711/161,162.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/12 11:08
S32	1689	707/204.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/12 11:08
S33	363	S31 AND S32	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/12 11:08
S34	2804	714/5,6.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/12 11:08
S35	6	S30 AND (S34 OR S33)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/12 12:42
S36	71	(bitmap) WITH synchronous$2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/12 12:42

EAST Search History

S37	6	S36 AND (S34 OR S33)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/13 09:17
S38	5561	714/5,6.ccls. OR 711/161,162.ccls. OR 707/204.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/13 09:18
S39	228	S38 AND log.AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/13 09:49
S40	299	S38 AND ((copied OR copy$3) NEAR (log$4))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/13 10:04
S41	84	S38 AND ((copied OR copy$3) NEAR (log OR logging OR logged))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/13 10:04
S42	5744	(transaction OR writ$4) NEAR log	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/21 15:44
S43	97	redo ADJ information	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/21 16:03
S44	2033	711/161,162.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/21 15:45

S45	3	S44 AND S43	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/21 15:45
S46	3046	bitmap.AB.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/22 15:00
S47	251	(bitmap NEAR (copy$3 OR copied))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/22 15:03
S48	1	(mirror WITH (bitmap NEAR (copy$3 OR copied)))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/22 15:01
S49	1792	711/162.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/22 15:03
S50	2033	711/161,162.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/22 15:03
S51	25	S50 AND S47	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/22 15:03
S52	270	((write ADJ ahead) OR ("write-ahead")) NEAR log$4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 14:49

S53	2044	711/161,162.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 14:57
S54	1786	707/204.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 14:57
S55	396	S53 AND S54	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 14:57
S56	3327	((synchronous$2 NEAR3 asynchronous$2) WITH (mode$1 OR switch$3))	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 14:57
S57	11	S56 AND S55	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 14:59
S58	2	"6591351".pn.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 14:59
S59	65	write ADJ logging	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 16:51
S60	1	remote NEAR (write ADJ log$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 16:51

S61	52	remote$2 WITH (write ADJ log$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 17:24
S62	1110	707/202.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 17:24
S63	14	S62 AND (remote NEAR log$4)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/12/28 17:24
S64	2	"20030159083"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/01/06 12:03

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L21	113	714/15.ccls.	US-PGPUB	OR	ON	2007/07/06 13:52
L22	77	mirror and redo and transaction	US-PGPUB	OR	ON	2007/07/06 13:52

Google

| mirroring data transaction redo   [Search]   Advanced Search
                                                         Preferences

---

Web   Results **1 - 10** of about **124,000** for <u>mirroring data transaction redo</u>. (0.20 seconds)

<u>MySQL AB :: MySQL Forums :: MaxDB :: Re: Backup/Restore via Split ...</u>
if the **redo** log don't match to the **data** volumes (think about the SAVEPOINT story), ...
Backup/Restore via Split **Mirror** techniques - How to accomplish it ...
forums.mysql.com/read.php?41,87274,87539 - 16k - <u>Cached</u> - <u>Similar pages</u>

<u>**Data** Guard & Remote **Mirroring**</u>
Better Network Efficiency - With Oracle Data Guard, only the **redo data** need to be sent to
the remote site. However, if a remote **mirroring** solution is used ...
www.oracle.com/technology/deploy/availability/htdocs/**Data**Guard**RemoteMirroring**.html -
69k - <u>Cached</u> - <u>Similar pages</u>

<u>**Data** Guard vs. Microsoft Database **Mirroring**</u>
Database **Mirroring** continuously sends a database's **transaction** log records to a .... Even
though a **Data** Guard physical standby cannot be open while **redo** is ...
www.oracle.com/technology/deploy/availability/htdocs/**Data**Guard**DatabaseMirroring**.html -
86k - <u>Cached</u> - <u>Similar pages</u>
[<u>More results from www.oracle.com</u>]

<u>**Mirror** Service - GigaSpaces 5.X Online Help - GigaSpaces Wiki Portal</u>
In the **redo** log the operations and **data** are stored as a Set of CacheBulk.BulkEntry objects,
similar to **transaction** objects (in case of an actual **transaction** ...
www.gigaspaces.com/wiki/display/GS/Mirror+Service - 45k - <u>Cached</u> - <u>Similar pages</u>

<u>High Availability: Frequently Asked Questions</u>
The principal and **mirror** servers must be separate instances of SQL Server 2005. In all
SQL Server databases, **data** changes are recorded in the **transaction** ...
www.microsoft.com/sql/technologies/highavailability/faq.mspx - 27k -
<u>Cached</u> - <u>Similar pages</u>

<u>DBAzine.com: An Oracle Instructor's Guide To Oracle **Data** Guard</u>
Because the systems are not **mirror** images of each other, **data** loss is also a ... The
primary database's LGWR process collects **transaction redo data** and ...
www.dbazine.com/oracle/or-articles/foot7 - 39k - <u>Cached</u> - <u>Similar pages</u>

<u>Cluster database with remote **data mirroring** - US Patent 6859811</u>
A method and apparatus for **mirroring data** between a plurality of sites is ... All
transactions in the **redo** log file that have a LSN that is less than or ...
www.patentstorm.us/patents/6859811-description.html - 74k - <u>Cached</u> - <u>Similar pages</u>

[PPT] <u>Oracle **Data** Guard - Technology Overview</u>
File Format: Microsoft Powerpoint - <u>View as HTML</u>
Redo Shipment. Risk of **Data** Loss. Protection Mode. Balance cost, availability,
performance, and **transaction** protection. Maximum Protection Mode ...
bigip-sage-back.oracle.com/presentations/ETC_**DATA**GUARD.ppt - <u>Similar pages</u>

<u>Recreation of archives at a disaster recovery site - Patent 6691139</u>
First, the parallel database servers 54, 56 fill up the online **redo** logs 58, 60 when the
transactions commit. Remote **mirroring** devices 66, 68 trap the disk ...
www.freepatentsonline.com/6691139.html - 43k - <u>Cached</u> - <u>Similar pages</u>

[PDF] Oracle database remote replication using Sun StorEdge **Data ...**
File Format: PDF/Adobe Acrobat - View as HTML
7.0 Oracle hot standby database vs. **mirroring** with Sun StorEdge **Data** ..... Oracle **Data**
Guard mirrors the online **transaction redo** log **data** from the source ...
www.sun.com/storagetek/white-papers/oracle**data**base_remotereplication.pdf -
Similar pages

Result Page: **1** 2 3 4 5 6 7 8 9 10   **Next**

Try Google Desktop: search your computer as easily as you search the web.
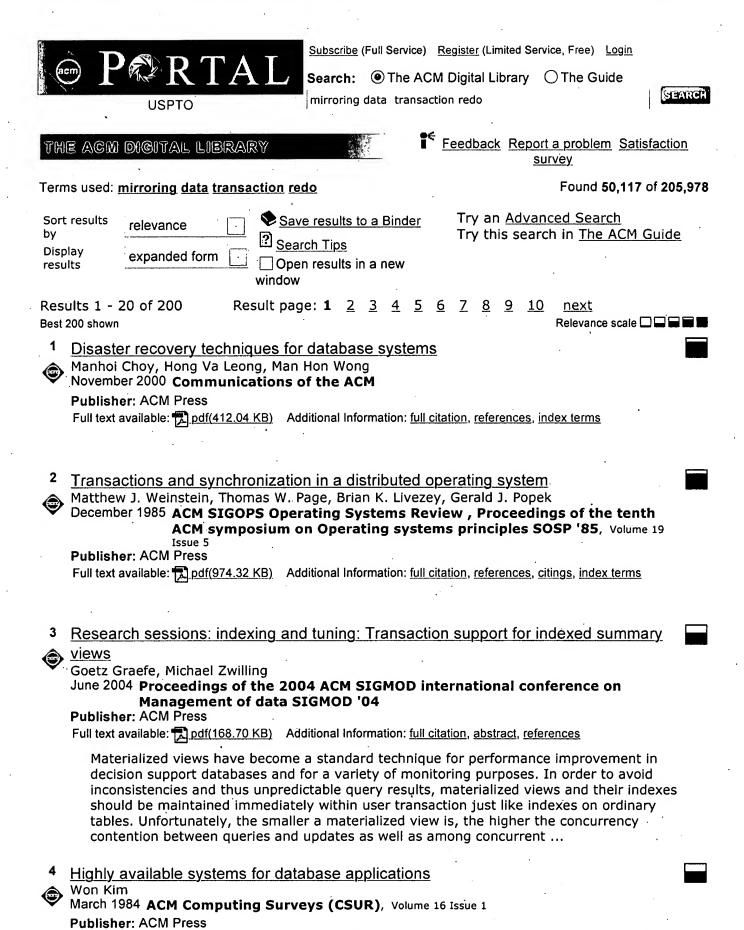
---

mirroring data transaction redo | Search |

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

---

Google Home - Advertising Programs - Business Solutions - About Google

©2007 Google

PORTAL
USPTO

Search: ⦿ The ACM Digital Library   ◯ The Guide

mirroring data transaction redo

SEARCH

THE ACM DIGITAL LIBRARY

ꙮ Feedback Report a problem Satisfaction survey

Terms used: **mirroring** **data** **transaction** **redo**   Found **50,117** of **205,978**

Sort results by   relevance ▢   ❧ Save results to a Binder   Try an Advanced Search
Display results   expanded form ▢   ⓘ Search Tips   Try this search in The ACM Guide
                                            ☐ Open results in a new window

Results 1 - 20 of 200   Result page: **1** 2 3 4 5 6 7 8 9 10   next
Best 200 shown   Relevance scale ☐▭▬◼◼

1 Disaster recovery techniques for database systems   ◼
   Manhoi Choy, Hong Va Leong, Man Hon Wong
   November 2000 **Communications of the ACM**

   Publisher: ACM Press
   Full text available: 🗎 pdf(412.04 KB)   Additional Information: full citation, references, index terms

2 Transactions and synchronization in a distributed operating system   ◼
   Matthew J. Weinstein, Thomas W. Page, Brian K. Livezey, Gerald J. Popek
   December 1985 **ACM SIGOPS Operating Systems Review , Proceedings of the tenth
                  ACM symposium on Operating systems principles SOSP '85**, Volume 19
                  Issue 5
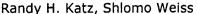   **Publisher:** ACM Press
   Full text available: 🗎 pdf(974.32 KB)   Additional Information: full citation, references, citings, index terms

3 Research sessions: indexing and tuning: Transaction support for indexed summary   ▬
   views
   Goetz Graefe, Michael Zwilling
   June 2004 **Proceedings of the 2004 ACM SIGMOD international conference on
              Management of data SIGMOD '04**
   Publisher: ACM Press
   Full text available: 🗎 pdf(168.70 KB)   Additional Information: full citation, abstract, references

   Materialized views have become a standard technique for performance improvement in
   decision support databases and for a variety of monitoring purposes. In order to avoid
   inconsistencies and thus unpredictable query results, materialized views and their indexes
   should be maintained immediately within user transaction just like indexes on ordinary
   tables. Unfortunately, the smaller a materialized view is, the higher the concurrency
   contention between queries and updates as well as among concurrent ...

4 Highly available systems for database applications   ▬
   Won Kim
   March 1984 **ACM Computing Surveys (CSUR)**, Volume 16 Issue 1
   Publisher: ACM Press

                              Additional Information: full citation, abstract, references, citings, index

Full text available: pdf(2.43 MB)

terms, review

As users entrust more and more of their applications to computer systems, the need for systems that are continuously operational (24 hours per day) has become even greater. This paper presents a survey and analysis of representative architectures and techniques that have been developed for constructing highly available systems for database applications. It then proposes a design of a distributed software subsystem that can serve as a unified framework for constructing database applica ...

5   Fast cluster failover using virtual memory-mapped communication

Yuanyuan Zhou, Peter M. Chen, Kai Li

May 1999 **Proceedings of the 13th international conference on Supercomputing ICS '99**

Publisher: ACM Press

Full text available: pdf(1.45 MB)   Additional Information: full citation, references, citings, index terms

6   Performance analysis of recovery techniques

Andreas Reuter

December 1984 **ACM Transactions on Database Systems (TODS)**, Volume 9 Issue 4

Publisher: ACM Press

Full text available: pdf(2.47 MB)   Additional Information: full citation, abstract, references, citings, index terms, review

Various logging and recovery techniques for centralized transaction-oriented database systems under performance aspects are described and discussed. The classification of functional principles that has been developed in a companion paper is used as a terminological basis. In the main sections, a set of analytic models is introduced and evaluated in order to compare the performance characteristics of nine different recovery techniques with respect to four key parameters and a set of other pa ...

7   Design transaction management

Randy H. Katz, Shlomo Weiss

June 1984 **Proceedings of the 21st conference on Design automation DAC '84**

Publisher: IEEE Press

Full text available: pdf(209.93 KB)   Additional Information: full citation, abstract, references, citings, index terms

A design transaction is a sequence of operations mapping a consistent version of an object into a new version. We describe a mechanism, based on version checkout and change files, that supports controlled sharing and is resilient to crashes in a network of workstations and database servers.

8   Implementing crash recovery in QuickStore: a performance study

Seth J. White, David J. DeWitt

May 1995 **ACM SIGMOD Record , Proceedings of the 1995 ACM SIGMOD international conference on Management of data SIGMOD '95**, Volume 24 Issue 2

Publisher: ACM Press

Full text available: pdf(1.67 MB)   Additional Information: full citation, abstract, references, citings, index terms

Implementing crash recovery in an Object-Oriented Database System (OODBMS) raises several challenging issues for performance that are not present in traditional DBMSs. These performance concerns result both from significant architectural differences between OODBMSs and traditional database systems and differences in OODBMS's target applications. This paper compares the performance of several alternative approaches to

implementing crash recovery in an OODBMS based on a client-server architecture. ...

9   MySQL Introduction
David Axmark, Michael Widenius
November 1999 **Linux Journal**
Publisher: Specialized Systems Consultants, Inc.
Full text available: 🖥 html(35.14 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>index terms</u>

   A look at the MySQL database--where it's been, where it is now, and where it's going

10 Evaluation of remote backup algorithms for transaction-processing systems
◆   Christos A. Polyzois, Héctor García-Molina
September 1994 **ACM Transactions on Database Systems (TODS)**, Volume 19 Issue 3
Publisher: ACM Press

Full text available: 📄 pdf(1.87 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>, <u>review</u>

   A remote backup is a copy of a primary database maintained at a geographically separate
   location and is used to increase data availability. Remote backup systems are typically
   log-based and can be classified into 2-safe and 1-safe, depending on whether transactions
   commit at both sites simultaneously or first commit at the primary and are later
   propagated to the backup. We have built an experimental database system on which we
   evaluated the performance of the epoch and the dependency recons ...

   Keywords: disaster recovery, hot spare, hot standby, remote backup

11 Recovery in the Calypso file system
◆   Murthy Devarakonda, Bill Kish, Ajay Mohindra
August 1996 **ACM Transactions on Computer Systems (TOCS)**, Volume 14 Issue 3
Publisher: ACM Press

Full text available: 📄 pdf(318.88 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>, <u>review</u>

   This article presents the deign and implementation of the recovery scheme in Calypso.
   Calypso is a cluster-optimized, distributed file system for UNIX clusters. As in Sprite and
   AFS, Calypso servers are stateful and scale well to a large number of clients. The recovery
   scheme in Calypso is nondisruptive, meaning that open files remain open, client modified
   data are saved, and in-flight operations are properly handled across server recover. The
   scheme uses distributed state amount the client ...

   Keywords: Calypso, cluster systems, distributed state, state reconstruction

12 Management of a remote backup copy for disaster recovery
◆   Richard P. King, Nagui Halim, Hector Garcia-Molina, Christos A. Polyzois
May 1991 **ACM Transactions on Database Systems (TODS)**, Volume 16 Issue 2
Publisher: ACM Press

Full text available: 📄 pdf(2.48 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>, <u>review</u>

   A remote backup database system tracks the state of a primary system, taking over
   transaction processing when disaster hits the primary site. The primary and backup sites
   are physically isolated so that failures at one site are unlikely to propogate to the other.
   For correctness, the execution schedule at the backup must be equivalent to that at the
   primary. When the primary and backup sites contain a single processor, it is easy to
   achieve this property. However, this is harder to do when ...

Keywords: database initialization, hot spare, hot standby, remote backup

13 A formal approach to undo operations in programming languages

George B. Leeman
January 1986 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 8 Issue 1
Publisher: ACM Press

Full text available: 🗎 pdf(2.74 MB)   Additional Information: full citation, abstract, references, citings, index terms

A framework is presented for adding a general Undo facility to programming languages. A discussion of relevant literature is provided to show that the idea of Undoing pervades several areas in computer science, and even other disciplines. A simple model of computation is introduced, and it is augmented with a minimal amount of additional structure needed for recovery and reversal. Two different interpretations of Undo are motivated with examples. Then, four primitives are defined in a langu ...

14 Empirical performance evaluation of concurrency and coherency control protocols for database sharing systems

Erhard Rahm
June 1993 **ACM Transactions on Database Systems (TODS)**, Volume 18 Issue 2
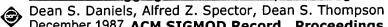**Publisher**: ACM Press

Full text available: 🗎 pdf(3.37 MB)   Additional Information: full citation, abstract, references, citings, index terms, review

Database Sharing (DB-sharing) refers to a general approach for building a distributed high performance transaction system. The nodes of a DB-sharing system are locally coupled via a high-speed interconnect and share a common database at the disk level. This is also known as a "shared disk" approach. We compare database sharing with the database partitioning (shared nothing) approach and discuss the functional DBMS components that require new and coordinated solutions for DB-shar ...

Keywords: coherency control, concurrency control, database partitioning, database sharing, performance analysis, shared disk, shared nothing, trace-driven simulation

15 Distributed logging for transaction processing

Dean S. Daniels, Alfred Z. Spector, Dean S. Thompson
December 1987 **ACM SIGMOD Record , Proceedings of the 1987 ACM SIGMOD international conference on Management of data SIGMOD '87**, Volume 16 Issue 3
Publisher: ACM Press

Full text available: 🗎 pdf(1.51 MB)   Additional Information: full citation, abstract, references, citings, index terms

Increased interest in using workstations and small processors for distributed transaction processing raises the question of how to implement the logs needed for transaction recovery. Although logs can be implemented with data written to duplexed disks on each processing node, this paper argues there are advantages if log data is written to multiple log server nodes. A simple analysis of expected logging loads leads to the conclusion that a high performance, microprocessor b ...

16 Storing a persistent transactional object heap on flash memory

Michal Spivak, Sivan Toledo
June 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN/SIGBED**

conference on Language, compilers and tool support for embedded systems LCTES '06, Volume 41 Issue 7

Publisher: ACM Press

Full text available: 📄 pdf(337.46 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>index terms</u>

We present the design and implementation of TinyStore, a persistent, transactional, garbage-collected memory-management system, designed to be called from the Java virtual machine of a Java Card. The system is designed for flash-based implementations of Java Card, a variant of the Java platform for smart cards. In the Java Card platform, objects are persistent by default. The platform supports transactions: a sequence of accesses to objects can be explicitly declared to constit ...

Keywords: nor flash, Java Card, flash, persistent heaps, persistent object stores, smart cards, transactions

17 Research session: architectural issues: C-store: a column-oriented DBMS

Mike Stonebraker, Daniel J. Abadi, Adam Batkin, Xuedong Chen, Mitch Cherniack, Miguel Ferreira, Edmond Lau, Amerson Lin, Sam Madden, Elizabeth O'Neil, Pat O'Neil, Alex Rasin, Nga Tran, Stan Zdonik

August 2005 **Proceedings of the 31st international conference on Very large data bases VLDB '05**

Publisher: VLDB Endowment

Full text available: 📄 pdf(210.85 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

This paper presents the design of a read-optimized relational DBMS that contrasts sharply with most current systems, which are write-optimized. Among the many differences in its design are: storage of data by column rather than by row, careful coding and packing of objects into storage including main memory during query processing, storing an overlapping collection of column-oriented projections, rather than the current fare of tables and indexes, a non-traditional implementation of transactions ...

18 The Kala basket: a semantic primitive unifying object transactions, access control, versions, and configurations

Sergui S. Simmel, Ivan Godard

November 1991 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages, and applications OOPSLA '91**, Volume 26 Issue 11

Publisher: ACM Press

Full text available: 📄 pdf(2.11 MB)   Additional Information: <u>full citation</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

19 A generator of direct manipulation office systems

Scott E. Hudson, Roger King

July 1986 **ACM Transactions on Information Systems (TOIS)**, Volume 4 Issue 2

Publisher: ACM Press

Full text available: 📄 pdf(2.45 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>, <u>review</u>

A system for generating direct manipulation office systems is described. In these systems, the user directly manipulates graphical representations of office entities instead of dealing with these entities abstractly through a command language or menu system. These systems employ a new semantic data model to describe office entities. New techniques based on attribute grammars and incremental attribute evaluation are used to implement this data model in an efficient manner. In addition, the s ...

20 <u>Persistent messages in local transactions</u>
David E. Lowell, Peter M. Chen
June 1998 **Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing PODC '98**
Publisher: ACM Press
Full text available: <u>pdf(976.77 KB)</u>   Additional Information: <u>full citation</u>, <u>references</u>, <u>index terms</u>

Results 1 - 20 of 200   Result page: **1** <u>2</u> <u>3</u> <u>4</u> <u>5</u> <u>6</u> <u>7</u> <u>8</u> <u>9</u> <u>10</u>   <u>next</u>